

IFES – Campus Serra

02/09/2019

Aluno: Ulysses Monte Alverne da Luz Carlos

Curso: BSI – 2019.2

Disciplina: Programação 1

Professor: Wagner Kirmse Caldas

Exercício individual II

Manual python turtle

o que é?

Turtle é uma biblioteca (isto é, um conjunto de comandos que pode ser adicionado à linguagem de forma opcional) da linguagem de programação python que tem por objetivo criar imagens, e trabalhar graficamente utilizando linhas de comandos.

Para que serve?

A capacidade de criar imagens utilizando comandos é muito útil para a criação de projetos e modelos que exigem precisão, pois como os comandos trabalham com parâmetros, tudo pode ser calculado friamente, e diferentemente do papel, não acontece de escorregar o lápis e um traço sair maior do que o outro quando eles precisam ser iguais.

Outra utilidade imensa do Python Turtle é a introdução dos conceitos iniciais de lógica de programação para iniciantes, pois ele oferece uma certa facilidade para criar formas geométricas e imagens simples, logo é fácil traçar um objetivo e buscar uma sequência de códigos que possam realizá-lo.

Como surgiu?

Na década de 1960 estavam no início as pesquisas sobre robôs e como poderíamos controlá-los, então utilizando as formas de representação gráficas da época foi criada a linguagem LOGO que poderia ser usada para programar os movimentos de um robô ou para fazer simulações em computadores onde poderiam ser traçadas possíveis formas de movimentação que poderiam ser úteis para robôs. Com o passar dos anos passou a ser utilizada para trabalhar programação com um público infantil e para programar os primeiros projetos de robótica escolares.

Basicamente o Python Turtle é uma forma de aproveitar as utilidades que o LOGO possuía na época, com a tecnologia das linguagens de programação modernas, e por isso funciona tão bem no Python, conhecida por ser uma linguagem versátil.

Como utilizar?

Para importar a biblioteca Turtle é preciso no início do programa fazer a chamada da mesma para que os seus comandos possam ser utilizados, e isso se dá da seguinte maneira:

```
from turtle import *
```

Depois, é necessário fazer algumas configurações básicas, pois por ser tão versátil, a linguagem oferece um certo grau de personalização quanto à velocidade e forma de representação, mas o mais comum é:

```
shape("turtle")  
speed(5)
```

Primeiros passos:

Para desenhar um quadrado nós temos o código:

```
from turtle import *  
shape("turtle")  
speed(5)
```

```
forward(100)
right(90)
forward(100)
right(90)
forward(100)
right(90)
forward(100)

done()
```

Onde “forward” faz a nossa tartaruga andar para frente, e o “100” que estamos usando como parâmetro é a quantidade de pixels que ela deve andar. Quando escrevemos “right” dizemos que ela deve virar à direita, quantificamos isso em graus, e como em um quadrado todos os vértices possuem 90°, é o parâmetro que usaremos em todas as curvas.

Comandos básicos:

from turtle import * - Isso diz para o Python que você quer à disposição todos os comandos da coleção de códigos que é a biblioteca Turtle.

shape(“turtle”) - sinaliza o formato que preferimos que tenha o “robô” que desenha, as opções são; “turtle” (tartaruga), “circle” (circulo), “triangle” (triângulo), “square” (quadrado), “arrow” (seta) ou “classic” (clássico). É importante destacar que essa mudança é apenas visual e não afeta o resultado.

speed(5) - É a função que diz ao robô em que velocidade ele deve andar, ela varia de 1(mais lenta) à 11 (mais rápida) .

forward(100) e **backward(100)** - Comando que sinaliza para a tartaruga andar e que recebe como parâmetro a quantidade de pixels, forward significa “para frente” e backward “para trás”.

left(45) e **right(45)** - Comando que diz para que lado a tartaruga deve virar (esquerda ou direita respectivamente) e quantifica isso em graus.

Color (“red”) e pencolor(“red”) - Permite mudar a cor da linha que será desenhada pela tartaruga, as cores mais simples podem ser escolhidas pelo nome, mas também é possível escolher por um código hexadecimal. Ex: #FF0000. A diferença entre os dois comandos é que pencolor não muda a cor do robô.

pensize(6) – Trata do tamanho da caneta, sua espessura.

Penup() - “Desencosta a caneta do papel” e permite a tartaruga andar sem riscar.

Pendown() - “Encosta novamente a caneta no papel e permite que o robô desenhe.

A vantagem de fazer parte do Python:

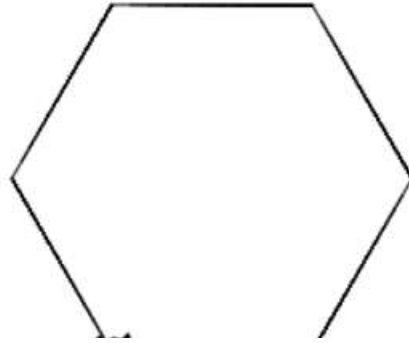
Python é uma linguagem completa e permite que suas estruturas convencionais sejam utilizadas para controlar a Turtle, o exemplo mais comum são as estruturas de loop, para fazer a tartaruga repetir algo muitas vezes.

Exemplos:

1° - polígono

```
from turtle import *  
shape("turtle")  
speed(5)
```

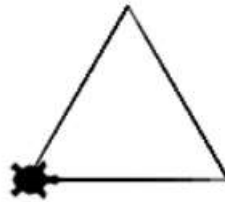
```
for i in range(6):  
    forward(100)  
    left(60)
```



2° - triângulo

```
from turtle import *  
shape("turtle")  
speed(5)
```

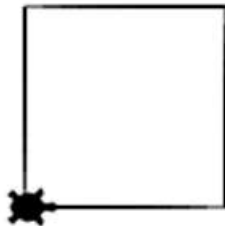
```
for i in range(3):  
    forward(100)  
    left(120)
```



3° - quadrado

```
from turtle import *  
shape("turtle")  
speed(5)
```

```
for i in range(4):  
    forward(100)  
    left(90)
```



4° - círculo

```
from turtle import *  
shape("turtle")  
speed(5)
```

```
for i in range(360):  
    forward(1)  
    left(1)
```



5° - função seno

```
from turtle import *  
shape("turtle")  
speed(11)
```

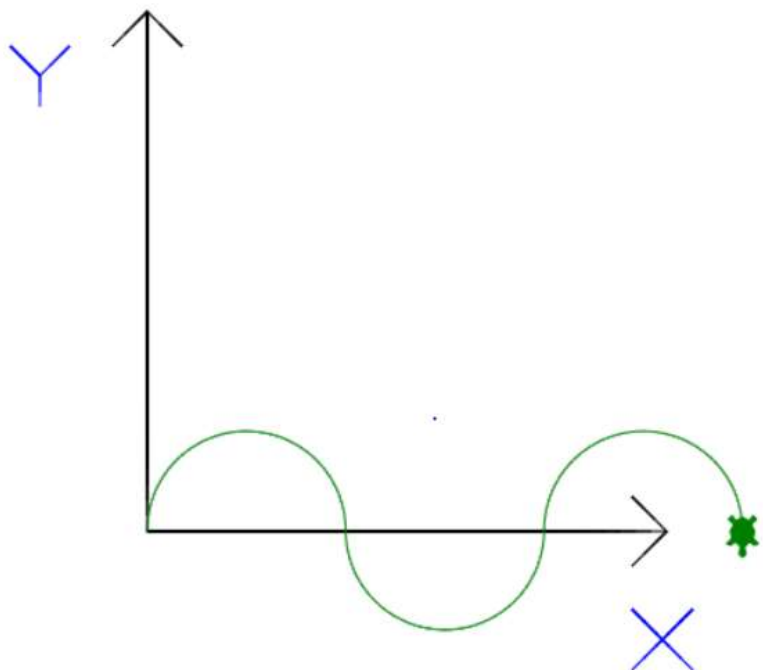
#desenhar gráfico

```
forward(300)  
right(135)  
forward(28.284)  
left(45)  
penup()  
forward(25)  
left(45)  
pendown()  
color("blue")  
forward(49.497)  
penup()  
left(135)  
forward(35)  
left(135)  
pendown()  
forward(49.497)  
color("black")  
penup()  
right(135)  
forward(35+25+20+20)  
right(135)  
pendown()  
forward(28.284)  
penup()  
right(180)  
forward(424.26-28.284)  
pendown()  
forward(28.284)  
left(90)  
forward(28.284)
```

```
right(45)  
penup()  
forward(25)  
left(45)  
pendown()  
color("blue")  
forward(24.042)  
right(90)  
forward(24.042)  
right(180)  
forward(24.042)  
right(45)  
forward(18)  
color("black")  
right(180)  
penup()  
forward(18+17+20)  
right(90)  
forward(17+25+20)  
pendown()  
right(90)  
forward(300)
```

#função seno

```
color("green")  
left(180)  
k=1  
for j in range(3):  
    for i in range(180):  
        forward(1)  
        right(k)  
        k=k*(-1)
```



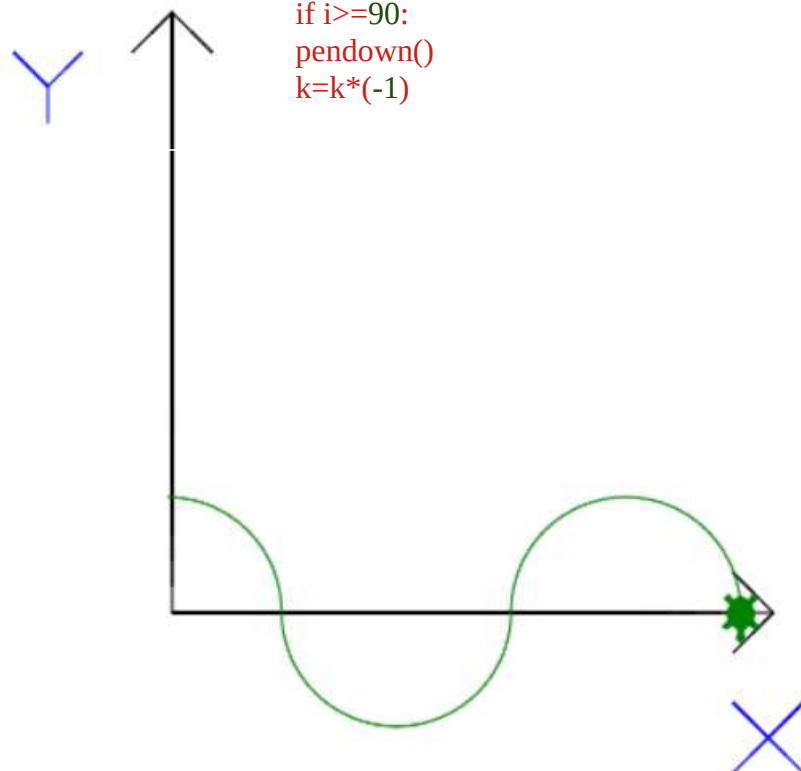
6º- Função Cosseno

```
from turtle import *  
shape("turtle")  
speed(11)
```

```
#desenhar gráfico  
forward(300)  
right(135)  
forward(28.284)  
left(45)  
penup()  
forward(25)  
left(45)  
pendown()  
color("blue")  
forward(49.497)  
penup()  
left(135)  
forward(35)  
left(135)  
pendown()  
forward(49.497)  
color("black")  
penup()  
right(135)  
forward(35+25+20+20)  
right(135)  
pendown()  
forward(28.284)  
penup()  
right(180)  
forward(424.26-28.284)  
pendown()  
forward(28.284)  
left(90)  
forward(28.284)  
right(45)
```

```
penup()  
forward(25)  
left(45)  
pendown()  
color("blue")  
forward(24.042)  
right(90)  
forward(24.042)  
right(180)  
forward(24.042)  
right(45)  
forward(18)  
color("black")  
right(180)  
penup()  
forward(18+17+20)  
right(90)  
forward(17+25+20)  
pendown()  
right(90)  
forward(300)
```

```
#função cosseno  
penup()  
color("green")  
right(90)  
forward(60)  
right(90)  
k=1  
for j in range (3):  
    for i in range(180):  
        forward(1)  
        right(k)  
        if i>=90:  
            pendown()  
            k=k*(-1)
```



7º- função quadrática

```
from turtle import *  
shape("turtle")  
speed(11)
```

#desenhar gráfico

```
forward(300)  
right(135)  
forward(28.284)  
left(45)  
penup()  
forward(25)  
left(45)  
pendown()  
color("blue")  
forward(49.497)  
penup()  
left(135)  
forward(35)  
left(135)  
pendown()  
forward(49.497)  
color("black")  
penup()  
right(135)  
forward(35+25+20+20)  
right(135)  
pendown()  
forward(28.284)  
penup()  
right(180)  
forward(424.26-28.284)  
pendown()  
forward(28.284)  
left(90)  
forward(28.284)  
right(45)  
penup()  
forward(25)  
left(45)  
pendown()  
color("blue")  
forward(24.042)  
right(90)  
forward(24.042)  
right(180)  
forward(24.042)  
right(45)  
forward(18)  
color("black")  
right(180)  
penup()
```

```
forward(18+17+20)  
right(90)  
forward(17+25+20)  
pendown()  
right(90)  
forward(300)
```

#função quadrática

```
color("green")  
speed(11)  
left(180)  
x=0  
a=1  
b=1  
c=1
```

```
for i in range(150):  
    y=(a*(x*x))+(b*x)+c  
    penup()  
    forward(y-1)  
    pendown()  
    forward(1)  
    penup()  
    backward(y)  
    right(90)  
    forward(1)  
    x+=0.1  
    left(90)
```

```
left(90)  
forward(x*10)  
x=0  
right(90)
```

```
for i in range(151):  
    y=(a*(x*x))+(b*x)+c  
    penup()  
    forward(y-1)  
    pendown()  
    forward(1)  
    penup()  
    backward(y)  
    right(-90)  
    forward(1)  
    x-=0.1  
    left(-90)
```

